Problem Set 5 **vgs3qt**

DATA STRUCTURES AND ALGORITHMS 2 - FALL 2023 DUE DECEMBER 5, 2023 AT 11:59 PM

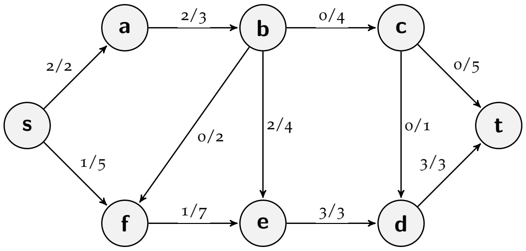
Collaboration Policy: You are encouraged to collaborate with up to 4 other students, but all work submitted must be your own *independently* written solution. List the computing ids of all of your collaborators in the `collabs` command at the top of the tex file. Do not share written notes, documents (including Google docs, Overleaf docs, discussion notes, PDFs), or code. Do not seek published or online solutions for any assignments. If you use any published or online resources (which may not include solutions) when completing this assignment, be sure to cite them. Do not submit a solution that you are unable to explain orally to a member of the course staff. Any solutions that share similar text/code will be considered in breach of this policy. Please refer to the syllabus for a complete description of the collaboration policy.

Collaborators: Yining Liu(yl2nr), Yuhao Niu(bhb9ba)

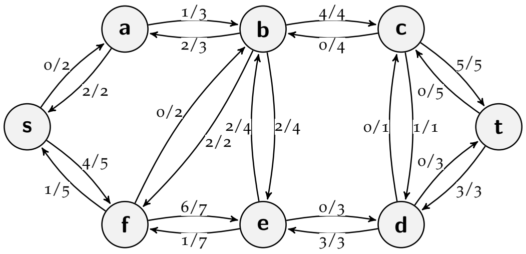
Sources: list your sources

PROBLEM 1 *Max Flow*

Given the following Flow Network G and Residual Graph G':



Flow Network G



Residual Graph G'

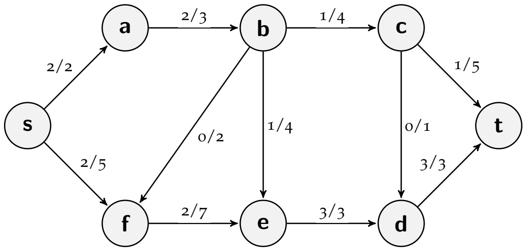
1. Find an augmenting path in the graph G'. List the nodes in the path you found in order (e.g., $s \to a \to b \to c \to d \to t$).

   Solution:

   $s \to f \to e \to b \to c \to t$

2. Update the Flow Network G above. You **must** edit the graph below (do not upload a picture).

   Solution:



3. Find the min cut of the graph. List the nodes on each side of the cut.

   Solution:

   $S = \{s, a, b, e, f\}$
   $T = \{c, d, t\}$

PROBLEM 2 *Element Uniqueness Reduction*

Reduce Element Uniqueness to Mode in $O(n)$ time. Element Uniqueness is defined as: given a list of numbers, return true if no number appears more than once (i.e., every number is distinct). Mode is defined as: given a list of numbers, return one of the numbers which appears most frequently in the list; i.e. if everything was unique it will return an arbitrary element.

Solution:

To reduce the Element Uniqueness problem to the Mode problem in $O(n)$ time, we will need the result of the Element Uniqueness problem when we solve the Mode problem.

Two problems will use the same map for inputs so we need to input the exact list of numbers and only take $O(1)$ for the next time.

Use an $O(n)$ time algorithm to find the mode of the list, counting the occurrences of each number, which allows us to find the mode in linear time. While counting the occurrences, check if any number occurs more than once. If all numbers occur exactly once (which means every count is 1), then the mode can be any number, and the Element Uniqueness problem would return true. If any number occurs more than once, the Element Uniqueness problem will return false. The overall time is $O(n)$.

Thus, we reduce Element Uniqueness to Mode in $O(n)$ time.

PROBLEM 3 *Reading and Evaluating Proofs*

Generative AI systems are exciting – *and scary*. They can answer many questions, but how much can we trust the results?

For this problem you will choose an generative AI system (e.g., ChatGPT, Bing (inside of Edge)) and ask it to do a proof for an algorithm we've studied in this unit—specifically, the proof that the reduction of Bi-Partite Matching to Max-Flow is correct. You'll then carefully read the proof it gives you and compare it to the version of that proof in our textbook, noting any issues or significant differences.

Here's a suggested prompt to give the system. You may use this unchanged, or alter it to try to get a better result.

Answer this question as if you were a computer scientist. Formally prove that the Bi-Partite Matching algorithm using a max-flow algorithm is correct (i.e, it always find the optimal matching between nodes in the bi-partite graph).

In your solution, provide the following; Give the name and version number of the generative AI system you've used.

Solution:

ChatGPT 3.5

In a sentence, describe the proof strategy used by the AI.

Solution:

ChatGPT approaches it with direct proof by first constructing the Max-Flow Graph and then prove the equivalence. It supposes there exists a maximum cardinality matching M in the original bipartite graph G.

Study the textbook's proof of this algorithm, Lemma 24.9 in Section 24.3 of the 4th edition of the textbook. In no more than 5 or 6 sentences, describe any issues or problems you see in the AI's result or how it differs from the textbook's proof. *Your answer might address the following questions: Do you think it successfully proves correctness? Are there gaps or odd logical jumps in the proof it provides? How different is it from the proof in the textbook? (If there are no issues to report, just say that.)*

Solution:

I think the AI failed to prove the correctness because it first simplified the proof. The proof from the book is much more rigorous and complete compared to AI's. For instance, when proving, the AI didn't clearly state that for any edge in a matching, we can find a corresponding flow in the network flow. With- out giving an instance for illustrating the proof, the AI lost the mathematical rigorousness and completeness in proving. Therefore, the AI's proof is incomplete.

Copy the prompt you gave the AI below.

Solution:

Proof that the reduction of Bi-Partite Matching to Max-Flow is correct.

Copy the AI's response (the proof) below.

Solution:

Claim: The Bipartite Matching problem can be reduced to an equivalent Max-Flow problem.
Proof:
Let's denote the bipartite graph as G = (U, V, E), where U and V are two disjoint sets of vertices, and E is the set of edges.

1. Construction of the Max-Flow Graph G′:

Create a new directed graph G′ = (V′, E′) as follows:

Add a source vertex s and connect it to all vertices in set U with edges of infinite capacity. Specifically, for each vertex u in U, add an edge (s, u) with capacity .

Add a sink vertex t and connect it to all vertices in set V with edges of infinite capacity. Specifically, for each vertex v in V, add an edge (v, t) with capacity .

For each edge (u, v) in the original bipartite graph G, add an edge from u in U to v in V′ with a capacity of 1. Specifically, add an edge (u, v) with capacity 1 for each edge (u, v) in E.

2. Proof of Equivalence:

Suppose there exists a maximum cardinality matching M in the original bipartite graph G. This means that there is no larger set of edges in G such that no two edges share a common vertex. We want to show that there exists a flow of capacity —M— from source s to sink t in G′.

Construct a flow in G′ as follows:

Set the flow along edges (u, v) (added in step 1) to 1 for each edge in the maximum matching M. Set the flow along all other edges in G′ to 0. It's easy to see that this flow is well-defined because it doesn't violate the capacity constraints (all edges have a capacity of 1), and it conserves flow at each vertex.

The value of this flow is —M— because we have set the flow of exactly one unit along each edge in the maximum matching M.

Conversely, suppose there exists a flow of capacity —M— from source s to sink t in G′. We want to show that there exists a maximum cardinality matching M in the original bipartite graph G.

Since the flow value is —M—, it implies that there are exactly —M— paths from s to t in G′.

Each path from s to t in G′ corresponds to an edge in the original bipartite graph G, as each path goes from a vertex in U to a vertex in V (paths can't cross from U to U or V to V because of the capacity constraints).

Therefore, the set of edges corresponding to these —M— paths in G is a maximum cardinality matching in G.

This completes the proof of the reduction from Bipartite Matching to Max-Flow. It shows that solving the Bipartite Matching problem is equivalent to finding the maximum flow in the constructed Max-Flow graph G′.

PROBLEM 4 *Algorithms and Society: Ethical and Social Issues*

In society the term *algorithms* is frequently used for what some call *algorithmic decision systems*. These are systems that rely on large amounts of data and algorithms that use AI or machine learning to make decisions in a wide range of important issues in society.

https://en.wikipedia.org/wiki/Automated_decision-making

While these systems are different in nature that almost all the algorithms topics we've studied in this course, some kind of algorithmic process is at the heart of such systems. It is appropriate for a computing student studying algorithms to be aware of this use of the term and to understand examples of such systems and the social and ethical challenges they pose.

Below is a list of articles, etc., that touch on algorithm-based systems and social or ethical issues in society. Choose one article that interests you, and answer the following questions.

1. List the title of the of the reading that you chose.

   Solution:

   Why colleges are using algorithms to determine financial aid levels.

2. In no more than five or six sentences, summarize how one algorithmic decision system discussed in the reading may lead to negative or undesirable consequences for individuals or societies. (The reading may discuss more than one, but you only need to write about one.)

   Solution:

   Designed to automate and streamline the process of enrolling individuals into programs or institutions, the algorithm can potentially lead to negative consequences because of biases embedded in its programming. If the algorithm is trained on historical data that contains biases, it can perpetuate or even exacerbate these biases. Furthermore, reliance on such systems can reduce human oversight and critical judgment, potentially overlooking unique circumstances or qualifications of individuals that do not fit the algorithm's parameters. This can lead to societal inequalities and a lack of fairness in access to opportunities, reinforcing existing social and economic divides.

3. In a few sentences, what actions do you think the computing personnel or organizations that create such systems could do to reduce possible negative or undesirable consequences? (Keep your answers brief!)

   Solution:

   The system can implement regular audits and bias checks on the algorithm to identify and correct any discriminatory patterns. Additionally, we can use diverse and inclusive training data to ensure the algorithm does not perpetuate existing biases.

Readings (choose one):

1. *Bias in AI-based models for medical applications: challenges and mitigation strategies*
   https://www.nature.com/articles/s41746-023-00858-z

2. *Machine Bias* (This is a longer article on risk assessments in criminal sentencing that got a lot of national attention.)
   `https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing`
3. *Why colleges are using algorithms to determine financial aid levels*
   `https://www.highereddive.com/news/colleges-enrollment-algorithms-aid-students/692601`
4. *Designing Ethical Self-Driving Cars*
   `https://hai.stanford.edu/news/designing-ethical-self-driving-cars`
5. *Algorithms that Run the World, an interview with Cathy O'Neil* (author of the book "Weapons of Math Destruction")
   `https://thedecisionlab.com/podcasts/algorithms-that-run-the-world-with-cathy-oneil`